DESIGN AND ANALYSIS OF ALGORITHMS

PROJECT REPORT ON

"EMERGENCY VEHICLE DISPATCHING SYSTEM"

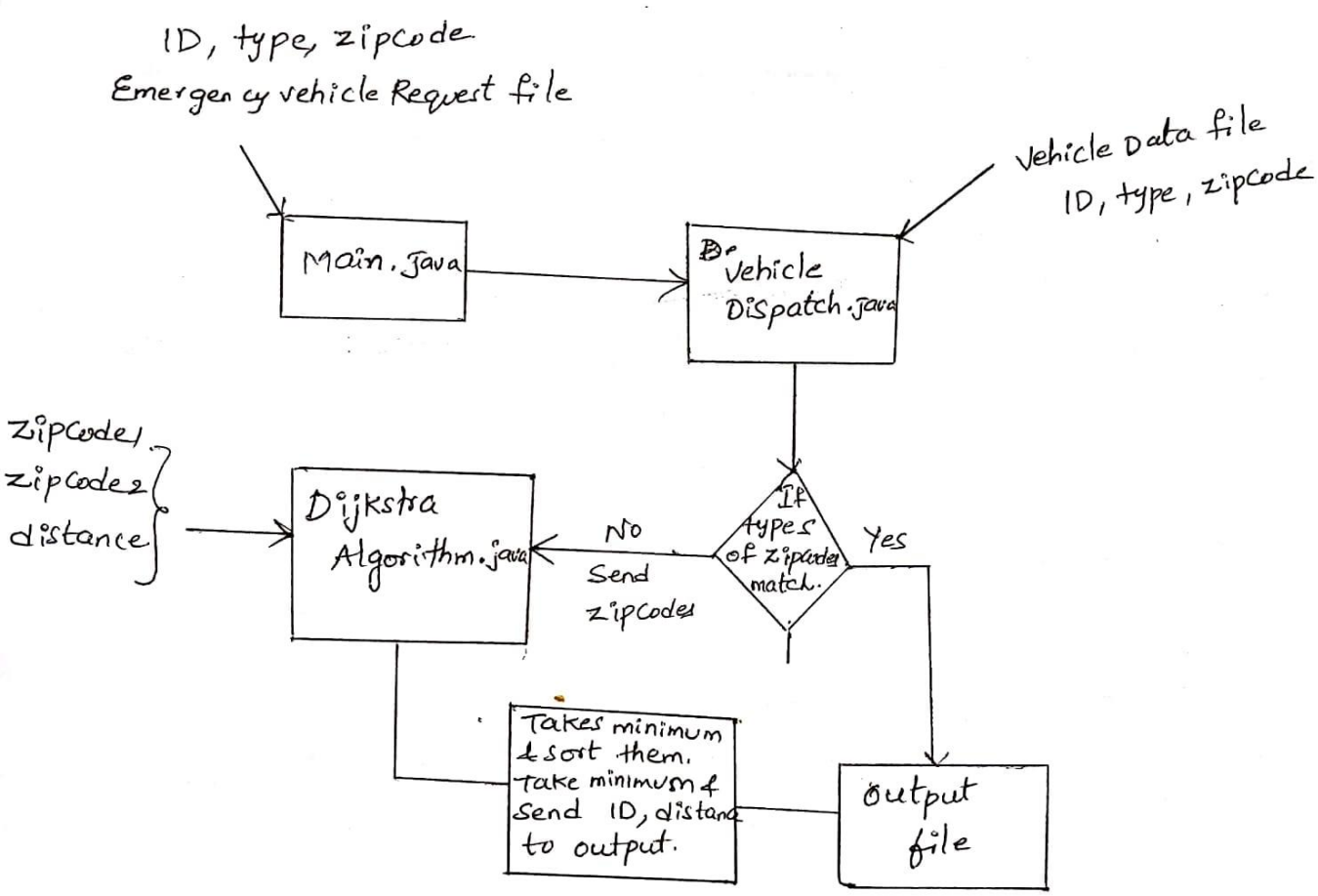
TEAM MEMBERS :

Fatema Hasta (16247903)
Vinay Jaibheem (16249942)
Sindhu Mudireddy (16242444)
Vamshi Thatikonda (16252921)
Sai Tejaswee Reddy Pasham (16233434)

PROJECT AIM -

To design an emergency vehicle dispatching system (given the three different types of emergency vehicles being Ambulance the first one, Fire Truck the second and Police car the third one), and to implement an algorithm that processes one request after the other to find the nearest available emergency vehicle.

FLOW DIAGRAM –



LANGUAGE USED – JAVA

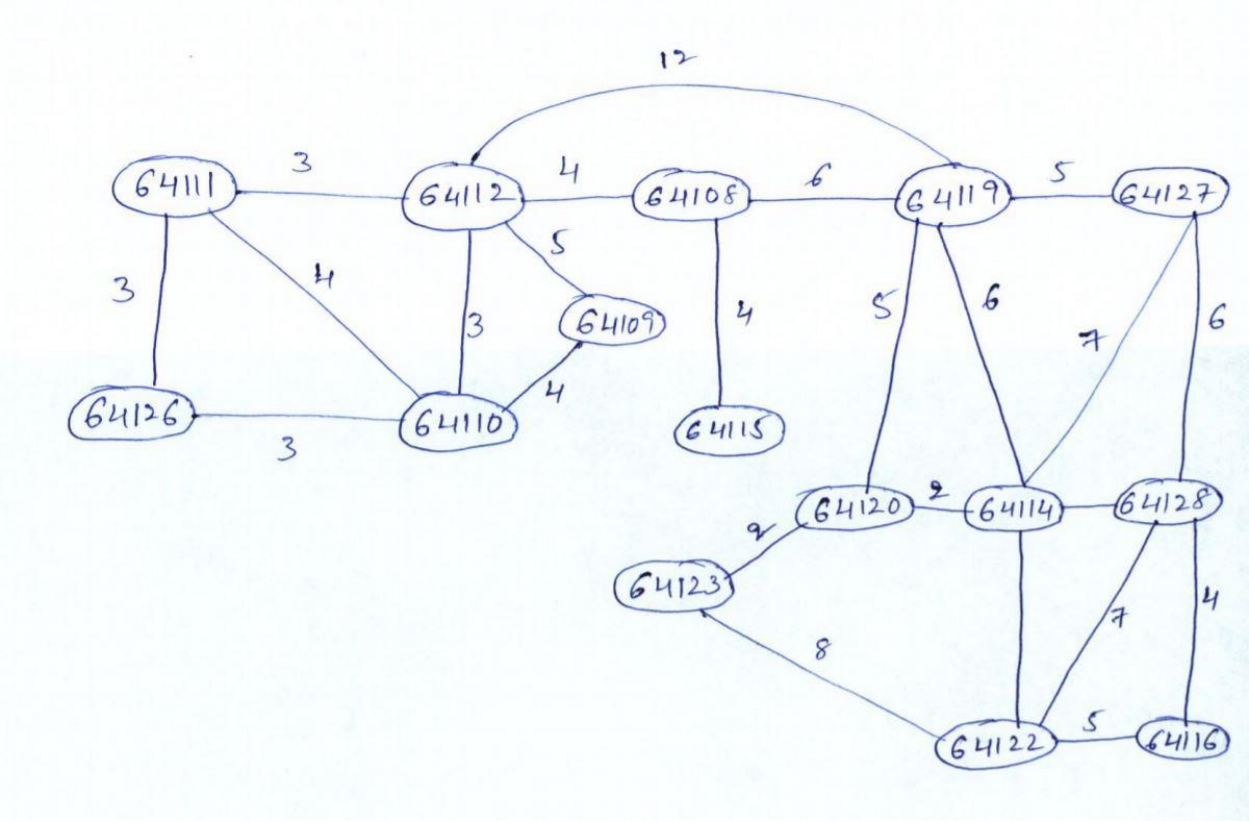
INTRODUCTION –

To design an emergency vehicle dispatching system, we initially started collecting and building the vehicle location, vehicle request data in separate text files. To utilize the above considered data, we generated few algorithms including Dijkstra's algorithm for vehicle requests, searching vehicles and dispatching of the vehicles. Thus, the shortest path between the required and the available zip codes of the vehicles is found using Dijkstra's algorithm and the output is taken out in a file.

ASSUMPTIONS CONSIDERED –

- We assumed that there are three types of vehicles Ambulance, Fire Truck and the Police car with the vehicle IDs being 1, 2 & 3 respectively.
- We also considered that every request needs only one emergency vehicle.
- We also assumed that the city we considered here has got zip codes ranging from 64108-64128.
- Here we assumed 64108 as 8, 64109 as 9 etc in a user-friendly manner.
- The graph of the city we assumed is directed and the distances between the zip codes are shown in the graph below.
- We assume that the vehicle present at the zip code is always available irrespective of the number of requests we get for that zip code.
- Here it is assumed that if the vehicles have got different zipcodes, then the vehicle from the same zipcode arrives.
- If in the request we get the type and zipcode both matched with the vehicle data, then we considered the distance as 0.

GRAPH –



Above is a undirected graph.
Here the numbers are assumed to be the zip codes – (vertices) and the three emergency vehicles are to be travelled from various locations distances between them are noted to be the edges.

ALGORITHMS USED –

Dijkstra's Algorithm – The purpose of this algorithm is to find the shortest path between the nodes in a graph. For example, if the nodes of the graph represent zip codes and edge path costs represent driving distances between pairs of zip codes connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one zip code and all other zip codes.
In this algorithm, we first start at the initial node which is set to zero and all the remaining nodes are to be set to infinity where we create few unvisited nodes package, and from the current node we consider all the edges and the respective distances are calculated. After this, the node is to be marked as visited, and is now

removed from the list of the unvisisted nodes and thus this process is therefore continued until all the edges are visited once and therefore the shortest path is thus finally attained.

ALGORITHM EFFICIENCY –

Here the overall time complexity of Dijkstra's Algorithm is taken as O(E).
Here we are storing the vertices in a Tresset and so our algorithm works on the basis of a treeset which gives us the time complexity of:

$$O(E + VlogV)$$

CONCLUSION –

Thus, the conclusion of the project is that we found the shortest path using the Dijkstra's Algorithm and also an algorithm is implemented that processes one request after the other and the nearest available emergency vehicle is found.

GITHUB LINK –

https://github.com/SaiTejasweeReddy/DaaProject_Spring2018

REFERENCES -

https://stackoverflow.com/questions/26547816/understanding-time-complexity-calculation-for-dijkstra-algorithm

https://www.javatpoint.com/java-hashmap

https://stackoverflow.com/questions/7465086/java-output-to-file-in-table-format

https://examples.javacodegeeks.com/core-java/nio/file-nio/path/java-nio-file-path-example/

https://stackoverflow.com/questions/29514554/linear-search-java-program

http://everythingcomputerscience.com/algorithms/Dijkstras_Algorithm.html

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#Using_a_priority_queue

https://www.quora.com/What-is-the-significance-of-using-a-priority-queue-in-Dijkstras-algorithm-What-difference-does-it-make-if-we-use-a-normal-queue